Utilizing Quartz Composer in Your Application

Interactive Motion Graphics for Mac OS X Tiger

Session 204

These are confidential sessions—please refrain from filming or taking pictures.

Graphics and Media

Utilizing Quartz Composer in Your Application

Interactive Motion Graphics for Mac OS X Tiger

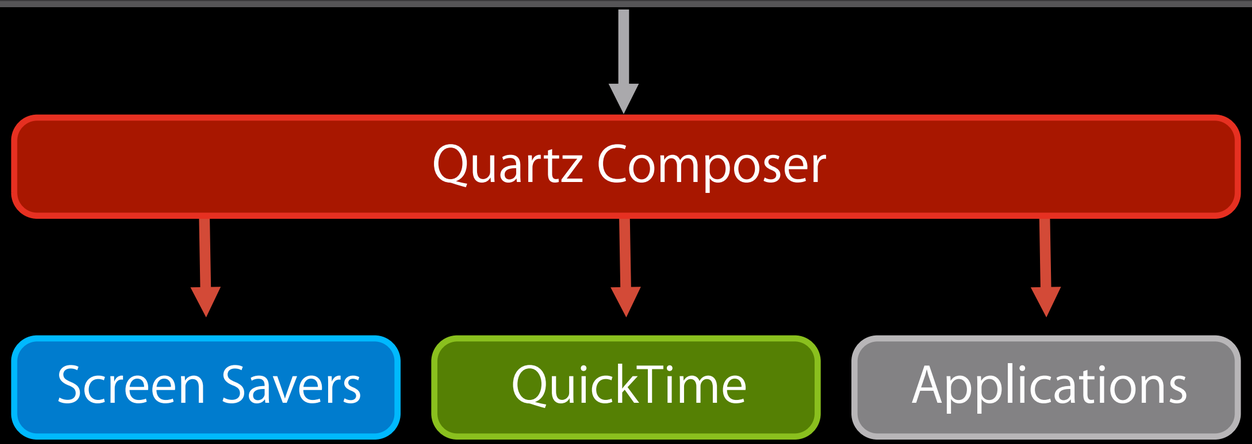
Pierre-Olivier Latour
Quartz Composer Architect

http://developer.apple.com/wwdc2005/

What Is Quartz Composer

- OS-level technology for real-time motion graphics
  - Designed around OpenGL for high performances and visual quality
  - Leverages many graphics and non-graphics technologies
- Creates “compositions” that render visuals
  - Support interactivity with various user devices
  - Can have input parameters and/or output results
- Comes with full-featured graphical editor (Dev. Tools)
- Supported by QuickTime (Tiger only)
- Easy to integrate into your applications

Quartz Composer Symbioses



“[Quartz Composer] is also the one thing that beats super glue for keeping eye candy nuts stuck to a Mac for hours on end.”

Tao of Mac, taoofmac.com

What You Will Learn

- Creating a parameterized composition with Quartz Composer
- Using compositions in Mac OS X Tiger
- Integrating Quartz Composer in your applications
- Building a Quartz Composer application from scratch

“I will give my left CENSORED and buy copies and evangelize for any mac vj developer who does the following:

Allow quartz composer to be used as a tool for designing custom effects in your program.”

DrEskaton, Internet forum

Tutorial #1

Creating a parameterized composition

http://developer.apple.com/wwdc2005/

Things to Remember

- Quartz Composer uses units and not pixels
  - Core Image filters are the exception as they manipulate image contents
- Organize the workspace with macros
  - Publish significant inputs and outputs
- Use the Splitter patch to group inputs
  - Set its type whenever possible
- Composition parameters
  - Create them by publishing inputs at the root macro patch level
  - Edit them in Viewer window

Tutorial #2

Using compositions in Mac OS X Tiger

http://developer.apple.com/wwdc2005/

QuickTime Support for Quartz Composer

- Mac OS X Tiger only
  - No user events
  - No internet downloads
  - No input parameters editing
- Quartz Composer movie track with type ‘qtz ‘
  - Progressive frames
  - Square pixels
  - Valid alpha channel contents
- Best performances with
  - QuickTime 7 Visual Context and Core Video APIs
  - QTKit for Cocoa applications
  - HIMovieView for Carbon applications

Quartz Composer Track Default Settings

- Default settings when importing
  - 640 x 480 pixels/30 seconds
- Change them in Terminal
  - Globally

```
defaults write NSGlobalDomain QuartzComposerDefaultMovieWidth 1024
defaults write NSGlobalDomain QuartzComposerDefaultMovieHeight 768
defaults write NSGlobalDomain QuartzComposerDefaultMovieDuration 300
```

  - Per application

```
defaults write com.apple.iMovie QuartzComposerDefaultMovieDuration 300
```

- Directly from Quartz Composer:
  - “Export as QuickTime Movie…” command

Customizing Track Settings From Your Application

- Using NSUserDefaults or CFPreferences:

```
NSUserDefaults* defaults = [NSUserDefaults standardUserDefaults];
[defaults setFloat:300.0 forKey:@"QuartzComposerDefaultMovieDuration”];
```

- Using QuickTime APIs
  - Dimensions with SetMovieMatrix() or SetMovieBox()
  - No easy way to change duration

Quartz Composer in Keynote

The QuickTime Backdoor

“[Live video input in QuickTime aware applications through Quartz Composer] kicks so much CENSORED , the CENSORED flew out the window.”

krezrock, VJForums.com

Things to Remember

- Compositions as screen savers in Library/Screen Savers
  - Input parameters become screen saver settings
- Native QuickTime support in Mac OS X Tiger
  - Simplest way to add motion graphics to existing QuickTime applications
  - Allow users to select ‘qtz’ files
  - Use QuickTime 7 APIs whenever possible

Tutorial #3

Integrating Quartz Composer in your applications

http://developer.apple.com/wwdc2005/

More QCView API

- Setting composition's input parameters

```
BOOL success =
[myQCView setValue:[NSNumber numberWithInt:5] forInputKey:@"In_Key"];
```

- Retrieving composition's output results

```
id foo = [myQCView valueForOutputKey:@"Out_Key"];
```

- Loading the composition programmatically

```
NSString* path = [[NSBundle mainBundle] pathForResource:@"My Composition"
ofType:@"qtz"];
BOOL success = [myQCView loadCompositionFromFile:path];
```

Parameters and Results Supported Value Types

Port Type	Value Class
Boolean, Index, Number	NSNumber or any object responding to -doubleValue, -floatValue or -intValue
Color	NSColor
String	String or any object responding to -description
Image	NSImage, CGImageRef, CIImage or CVImageBuffer
Structure	NSArray or NSDictionary

Things to Remember

- Quartz Composer rocks
- Quartz Composer rocks
- Quartz Composer rocks

Things to *Really* Remember

- Quartz Composer is for Quartz Extreme machines only

```
boolean_t valid = CGDisplayUsesOpenGLAcceleration(kCGDirectMainDisplay);
```

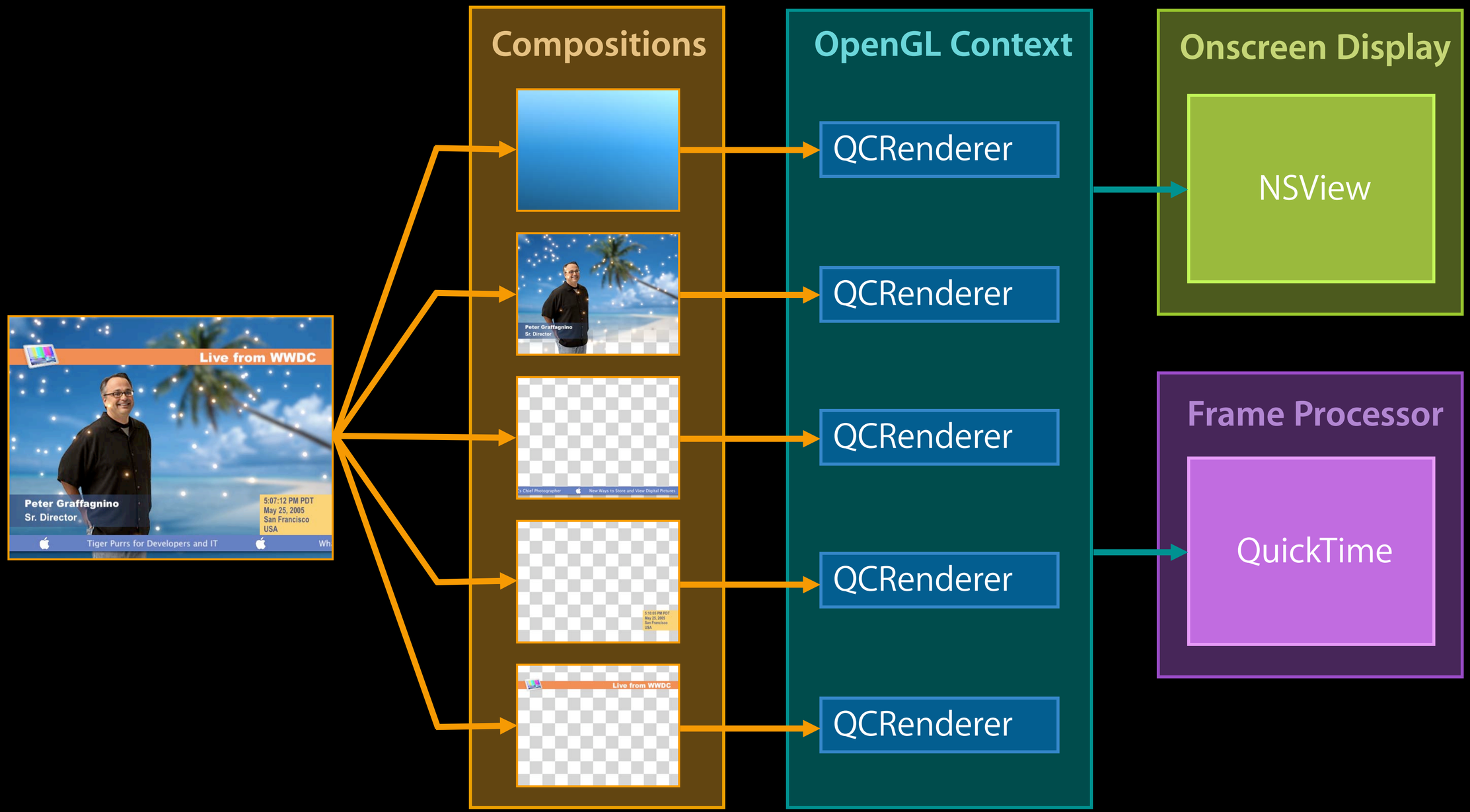
- Communicating with compositions from host application is easy
  - Move outside of the composition complex operations
- Composition inputs and outputs are identified by keys not names
  - Keys are built automatically frompublished portsinitial name
  - Use tooltips in editor to see published port keys
- Don’t forget to
  - Add QuartzComposer.palette to Interface Builder
  - Include Quartz.framework in Xcode project

Tutorial #4

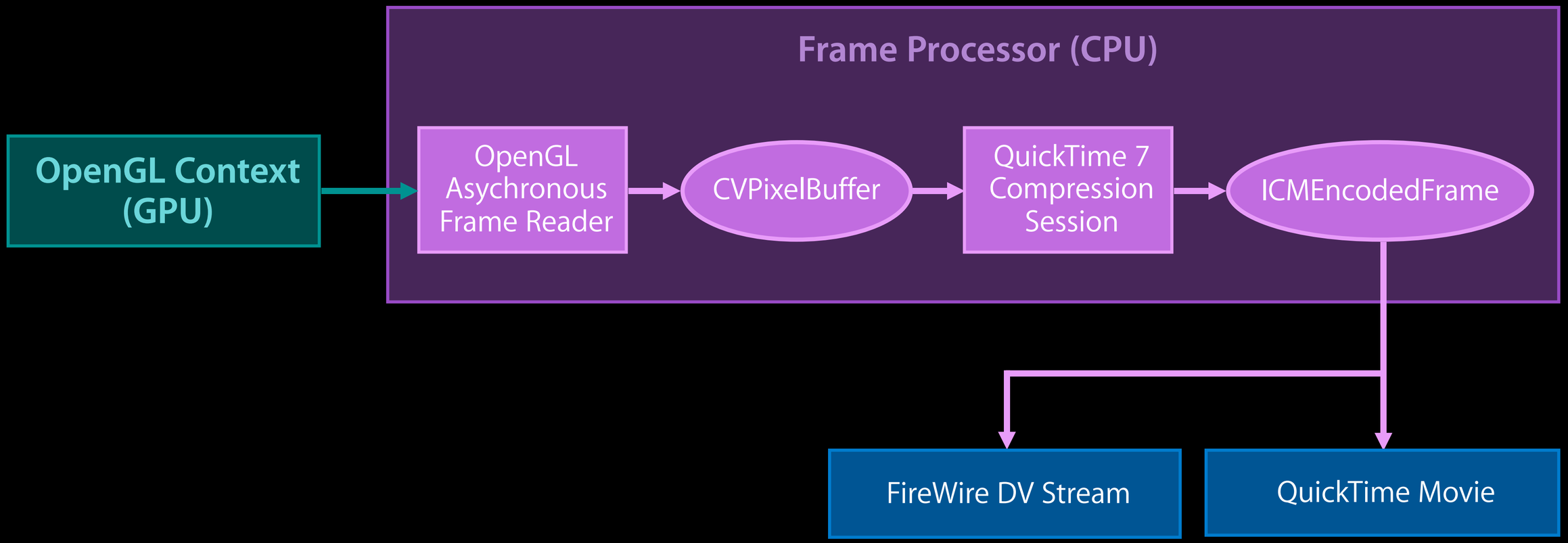
Building a Quartz Composer application from scratch

http://developer.apple.com/wwdc2005/

QCTV Architecture



Frame Processing in Detail



Things to Remember

- Prototype in Quartz Composer editor
- Consider design choices
  - Monolithic versus separate compositions
  - Features distribution between host application and compositions
- Composite compositions using QCRenderers in OpenGL context
  - Don’t forget to call -[NSOpenGLContext flushBuffer]
  - OpenGL viewport defines the QCRenderer rendering area
  - QCRenderer saves / restores OpenGL states except GL_CURRENT_BIT
- Possible to build UI on-the-fly for composition input parameters
- Speed up development for simple cases using QCView/QCPatchController and Cocoa bindings

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Creating Quartz Composer Compositions Lab	Tiger Lab Tuesday 3:30PM
Core Video and QuickTime Video Pipelines Lab	Graphics and Media Lab Wednesday 12:00PM
Core Image, Image Units, Image I/O, ColorSync, and Image Capture Lab	Graphics and Media Lab Thursday 12:00PM
OpenGL and Quartz Composer Lab	Graphics and Media Lab Friday 9:00AM
Feedback Forum: Graphics and Media	North Beach Friday 2:00PM

Who to Contact

Travis Brown
Graphics & Imaging Evangelist
travis@apple.com

Pierre-Olivier Latour
Quartz Composer Architect
pol@apple.com

Q&A

Travis Brown
Graphics & Imaging Evangelist
travis@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference